ISBN: 978-3-95450-231-8 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2023-THPL017

AUTOMATING BEAM DUMP FAILURE DETECTION USING COMPUTER VISION

F. Huhn*, F. M. Velotti, B. Goddard, V. Bencini, CERN, Geneva, Switzerland

Abstract

The CERN SPS Beam Dump System (SBDS) is responsible for disposing the beam in the SPS in case of any machine malfunctioning or end of cycled operation. This is achieved by the actuation of kicker magnets with predefined pulses, which aim to: i) deviate the beam towards the absorber block (TIDVG); ii) dilute the particle density. Evidently, a malfunction of this system may have negative consequences, such as the absorber block degrading if the beam is not sufficiently diluted, unwanted activation of the surroundings or even damage to the vacuum chamber in case of complete failure. By leveraging a combination of real images from a beam screen device and data from simulations, we train an online monitoring system to identify potential failures of the SBDS from real-time images. This work improves the safety of the operation of the SPS and contributes towards the goal of automating the operation of accelerators.

INTRODUCTION

The Super Proton Synchrotron (SPS) is part of the CERN accelerator chain, receiving beam from the Proton Synchrotron and feeding fixed target experiments and the Large Hadron Collider (LHC). The SPS features a beam dump system, the SPS Beam Dump System (SBDS) [1, 2], which is responsible for the disposal of the beam in case of machine malfunctioning or end of cycled operation. The SBDS is composed of three horizontal (MKDH), three vertical (MKDV) kickers and an absorber block (TIVDG), as shown in Fig. 1. These kickers are actuated to deflect the beam towards the TIDVG. Moreover, they are pulsed such that the beam is diluted on the TIVDG – by continuously changing the point where the beam hits the TIDVG, particle density is reduced, avoiding damage to the absorber block. Therefore, the SBDS is a critical component of the SPS and it follows that its monitoring is important for the safe operation of the accelerator. Thus, we develop a tool based on a Convolutional Autoencoder, widely used in computer vision and machine learning, to detect failures in the SBDS from BTV [3] images. Importantly, it can detect both failures on which it was trained on and failures not seen during training. It can also predict certain physical parameters of the beam, providing additional information to the operator.

BTV IMAGES

A BTV is located 3665 mm in front of the TIVDG (not depicted in Fig. 1), through which the beam passes before hitting the TIDVG. The BTV is oriented at an angle of 45° to the beamline, allowing for its image to be acquired from the side [5]. An image is acquired and stored for each beam dump. The left panel of Fig. 2 shows an example of a BTV acquisition (after preprocessing) of a normal SFTPRO (a particular beam type that fills completely the SPS circumference) dump.

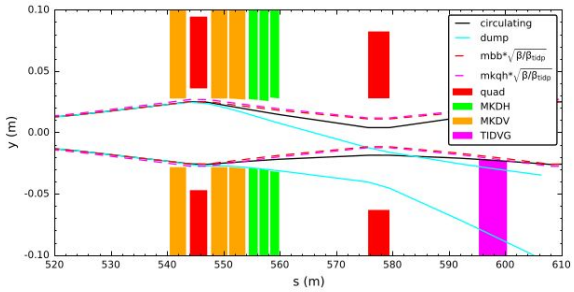

Figure 1: Schematic of the SBDS [4].

Images of the BTV can also be obtained from simulations based on particle tracking. An example of a BTV image obtained from a simulation is shown in the right panel of Fig. 2. The agreement between the simulation and the real BTV is reasonably good, though there are visible differences, such as the outline of the beam. In these simulations, various parameters of the beam, such as emittance, number of bunches, number of batches, batch spacing, etc., can be changed. This makes these simulations very appealing, as they can be used to generate a labeled dataset (image and parameters) that covers a wide range of combinations, which would be extremely difficult to do in the physical machine.

MODEL ARCHITECTURE

Autoencoder and Reconstruction

As mentioned above, we wish to detect failures in the SBDS from BTV images. Moreover, we wish to predict physical parameters of the beam. Thus, our problem is that of a regression from BTV images to a combination of numerical (physical parameters) and categorical (failures) variables. A regressor, however, would generally not be able to identify failures on which it was not trained. This is an

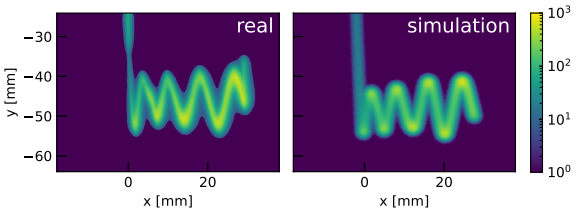

Figure 2: BTV image of an SFTPRO beam. Denoised acquisition (left) and simulation (right). Arbitrary pixel units.

* francisco.huhn@cern.ch

ISBN: 978-3-95450-231-8 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2023-THPL017

important issue because our monitoring system should be robust to many types of failures, even those not previously observed. We overcome this by employing autoencoders [6], which have found numerous applications, such as dimensionality reduction [7], image denoising [8], and, most importantly to this work, anomaly detection [9, 10]. Autoencoders consist of two submodels, an encoder and a decoder. The encoder transforms a high-dimensional input into a lower-dimensional output, latent vector (or code), which encodes the information of the input. Then, the decoder takes the latent vector and attempts to reconstruct the original input. In our particular application, we train the autoencoder such that the latent variables match the numerical and categorical variables (physical parameters and failures) that we wish to predict, thus making the encoder a regressor. Notwithstanding, we also allow the inclusion of free latent variables as in common autoencoders to take into account features other than the physical parameters we wish to predict. The decoder then takes this mixed vector of predicted physical parameters and general features of the input and tries to reproduce the original BTV image given as input.

It is the reconstruction that enables the detection of previously unseen failures[1]. If the image of such a failure is fed to the autoencoder, the encoder will make a (not necessarily correct) prediction of the physical parameters, exactly as it would do for an image of a normal beam. Therefore, it would be difficult to flag the failure purely based on these values. However, because the decoder was trained to reconstruct the images of normal operation and failures present in the training set (which can be flagged by the latent variables), its reconstruction of the normal operation would be good, whereas that of the unseen failure would be poor. Therefore, by comparing the reconstruction to the original BTV image, i.e. computing the reconstruction error, we can potentially detect failures not present in the training set.

Hidden Layers

Since our inputs are images, we employ Convolutional Neural Networks (CNN) [11], widely used in computer vision. The encoder contains five layers, the first four convolutional and the last fully connected. Each of the convolutional layers is followed by batch normalisation [12] and activation with leaky rectified linear units (ReLUs). For categorical latent variables, a sigmoid is applied to transform the output of the layers to probabilities. The decoder follows the same architecture in reverse and with Transposed CNNs [11], followed by the application of softmax to transform the output to a probability distribution. This is important for the computation of the reconstruction loss as detailed in the next section. In both CNNs and Transposed CNNs, the kernel size is 3, the stride is 2 and padding is 1. The hidden dimensions are 32, 64, 128 and 256 at the final convolutional layer of the encoder, with the order reversed for the decoder.

[1] Assuming such a failure somehow affects the BTV image.

TRAINING AND TESTING

Dataset

In this work, we restrict ourselves to simulated data. The transfer of this tool to real data is also under development. The dataset consists of 1024 samples and is generated according to the distributions shown in Table 1. The numerical variables are: jitter_x and jitter_y, corresponding to small horizontal and vertical shifts that can happen in the machine; ε_x and ε_y, the horizontal and vertical normalised emittances. Finally, mkdv_fail is the sole categorical variable and indicates whether an MKDV has failed. Failure, here, means the MKDV does not produce a kick. This parameter is Bernoulli-distributed with parameter 0.1, i.e. an MKDV fails on 10% of samples on average. Each sample is a pair of the BTV image and the vector of the physical parameters of one simulation.

The training set consists of 896 randomly selected samples from the 1024 and the test set consists of the remaining 128 plus 16 new samples corresponding to MKDH failures. These failures are not present in the training set and serve to test if the model can detect previously unseen failures.

Table 1: Parameter Distribution of Dataset

Parameter	Distribution	Unit
jitter_x	$\mathcal{N}(0, 3^2)$	mm
jitter_y	$\mathcal{N}(0, 3^2)$	mm
ε_x	$\mathcal{N}(5, 1^2)$	10^{-6} mm mrad
ε_y	$\mathcal{N}(3.5, 1^2)$	10^{-6} mm mrad
mkdv_fail	$\mathcal{B}(0.1)$	

Training

The numerical variables are normalised to follow a standard normal distribution and mkdv_fail is binary. The images are transformed to a resolution of 128x64 and subsequently normalised such that the sum of the pixels is 1, which allows the use of probability-based losses, as explained next.

The loss function has three components. First, mean squared error (MSE) is applied to the numerical latent variables. Second, cross entropy is applied to mkdv_fail. Finally, we apply the Kullback-Leibler divergence (KL divergence) for the image reconstruction loss. The intensities of the pixels of BTV images span across multiple orders of magnitude, from background noise away from the beam to very high intensities at its peaks and troughs. Thus, an application of usual metrics, such as MSE, would result in a bad reconstruction of all pixels but those of the very highest intensities, as these would dominate the loss. Because the KL divergence is an entropy-based metric, it works in log-probabilities and does not suffer from this issue. It regularises the losses of pixels of different orders of magnitude, thus making it more amenable to a better reconstruction. Indeed, we have found substantial improvements in reconstruction quality using this loss when compared to MSE.

ISBN: 978-3-95450-231-8 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2023-THPL017

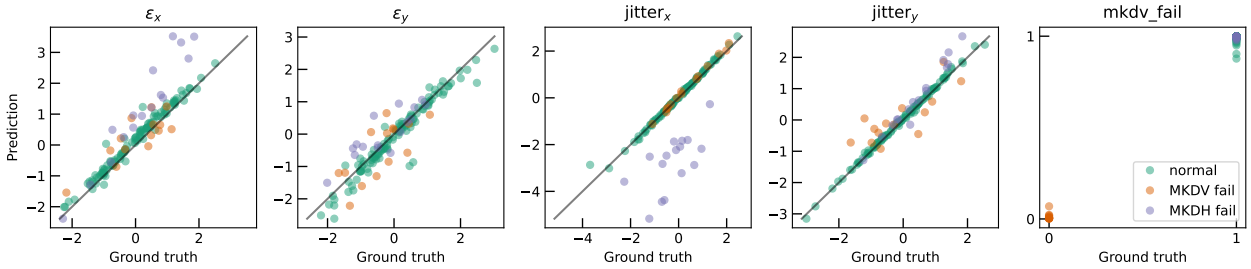

Figure 3: Predictions of the physical parameters on test set.

We use the Adam optimiser with default parameters [13] on batches of 128 samples, over 100 epochs, with a learning rate decay of $1/2$ every 10 epochs.

RESULTS

Figure 3 shows the predictions of the physical parameters on the test set. For the types of cases seen in training, normal operation and MKDV failure, there is very good agreement between predictions and ground truth. In particular, the model can identify an MKDV failure with 100% recall. These cases, however, show a slight dispersion in the other variables, especially in vertical jitter and emittance. This can be explained by the fact that an MKDV failures results: in a vertical shift, i.e. the same effect as vertical jitter; and in a vertical compression, which increases particle density, similarly to a lower-emittance beam. Unsurprisingly, the predictions of the MKDH failure cases are poor, which is expected, as the model was not trained on these samples. Here, as described previously, we use the reconstructions to assess whether this failure occurred.

Figure 4 shows the original BTV image, its reconstruction and a histogram of its pixels for an example of each of the three types of cases. It is clear that the model reconstructs well the two types in the training set, with a slightly larger error for the MKDV failure case. This is possibly due to the low number of training samples. Furthermore, the visual assessment is supported by the histograms, which show great agreement between original and reconstruction at the most important orders of magnitude. On the other hand, the MKDH failure is badly reproduced, resulting in high reconstruction error, with noticeable difference in the histograms, especially the rightmost bucket.

Figure 5 shows the distribution of reconstruction error. In general, MKDH failure cases show higher error than the other two types. Therefore, a threshold criterion, such as reconstruction loss $> 4 \times 10^{-1}$ (red dashed line), could be used to flag a potential failure. Or, at least, a type of case not covered by the training set, for which predictions are not to be trusted. However, there are two non-MKDH-failure samples that would be flagged, i.e. false positives. A slightly larger training set and tuning of the model would possibly eliminate these. Overall, this criterion results in 98.6% accuracy, 87.5% precision and 100% recall on the test set.

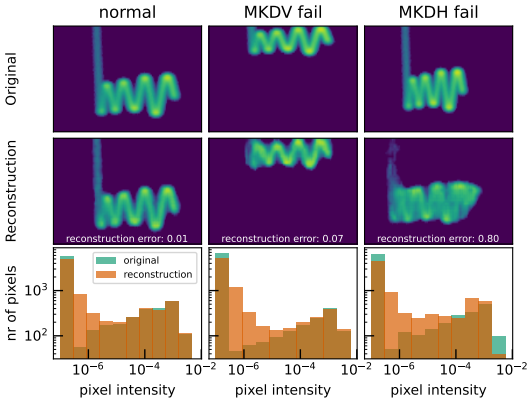

Figure 4: Original image, its reconstruction and histogram of pixel intensity, for each type of case.

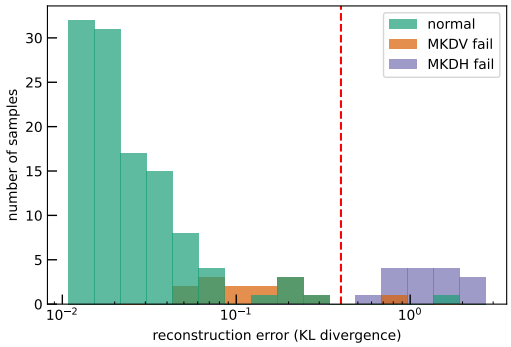

Figure 5: Reconstruction error.

CONCLUSION

We trained a model that infers beam parameters and detects failures with great accuracy from dump BTV images alone. Importantly, the model is robust, detecting not only failures present in the training set, but also previously unseen failures. Its performance can likely be improved with a larger dataset and hyperparameter optimisation. Further development and research is currently undergoing in: i) hyperparameter tuning of the model; ii) real data testing; iii) mixed dataset training, i.e. with both simulated and real data; iv) generalisation (e.g. to other beam types). This will contribute to the improvement and eventual deployment of the model in the operation of the SPS.

ISBN: 978-3-95450-231-8 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2023-THPL017

REFERENCES

[1] F. M. Velotti *et al.*, "Feasibility Study of a New SPS Beam Dump System," in *Proc. IPAC'15*, Richmond, VA, USA, May 2015, pp. 3930–3933. doi:10.18429/JACoW-IPAC2015-THPF097

[2] P. V. Trappen *et al.*, "SPS Beam Dump System (SBDS) Commissioning After Relocation and Upgrade," in *Proc. IPAC'22*, Bangkok, Thailand, 2022, paper THPOST039, pp. 2530–2532. doi:10.18429/JACoW-IPAC2022-THPOST039

[3] E. Bravin, S. Burger, G. Ferioli, G. J. Focker, A. Guerrero, and R. MacCaferri, "A new TV beam observation system for CERN," 2005. https://cds.cern.ch/record/895159

[4] F. M. Velotti, "REMOTE: Applications of computer vision and forecasting to the CERN accelerators," 2022. https://cds.cern.ch/record/2808782

[5] S. Burger, E. Bravin, F. Roncarolo, A. Topaloudis, F. Velotti, and E. Veyrunes, "New CERN SPS Beam Dump Imaging System," in *Proc. IBIC'21*, Pohang, Rep. of Korea, 2021, paper TUPP22, pp. 254–258. doi:10.18429/JACoW-IBIC2021-TUPP22

[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[7] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006. doi:10.1126/science.1127647

[8] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th International Conference on Machine Learning*, 2008, pp. 1096–1103. doi:10.1145/1390156.1390294

[9] B. Thompson, R. Marks, J. Choi, M. El-Sharkawi, M.-Y. Huang, and C. Bunje, "Implicit learning in autoencoder novelty assessment," in *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No.02CH37290)*, vol. 3, 2002, 2878–2883 vol.3. doi:10.1109/IJCNN.2002.1007605

[10] M. Sakurada and T. Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction," in *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, 2014, pp. 4–11. doi:10.1145/2689746.2689747

[11] V. Dumoulin and F. Visin, *A guide to convolution arithmetic for deep learning*, 2018.

[12] S. Ioffe and C. Szegedy, *Batch normalization: Accelerating deep network training by reducing internal covariate shift*, 2015.

[13] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 2017.